COMP 110-001
Review of Chapter 1 & 2

Yi Hong

May 18, 2015

Today

- Review of Chapter 1 & 2
- Review programs in lectures & labs

Hardware vs. Software

- Hardware - physical machine
  - CPU, Memory
- Software - programs that give instructions to the computer
  - Windows XP, Games, Eclipse

Measuring Data

- 1 bit (binary digit): 0 or 1
- 1 byte: 8 bits
  - 00000000 ~ 11111111
  - 2^8 = 256 possible states
- An example of a byte
  - 0 1 0 1 0 0 1 0
  - As decimal number: 82 = 2^1 + 2^4 + 2^6

Measuring Data

- 4 bytes: 4 * 8 = 32 bits
  - 2^32 possible states
- Size of int type in Java
- If we use 4 bytes to represent an integer, what is the range?
  - Unsigned: 0 ~ 2^32-1 (starts from 0)
  - Signed: -2^31 ~ 2^31-1

Primitive Types

- Examples (For a full list, check p. 52)

Type	Size	Example	Remarks
int	4 bytes	3443, -1024	Integer only Smaller range +, - , * , / , %
double	8 bytes	-0.4, 3.2, 3.343*10^100	Much larger range Limited precision +, - , * , / , %
boolean	1 bit	true, false	And or negation && || !
char	2 bytes	‘a’, ‘0’, ‘-’, ‘%’	Single quotes

Variables

- Container of Data
  - Data can be of Class type or Primitive type
- Declaration of Variable:

type *variable_name*;

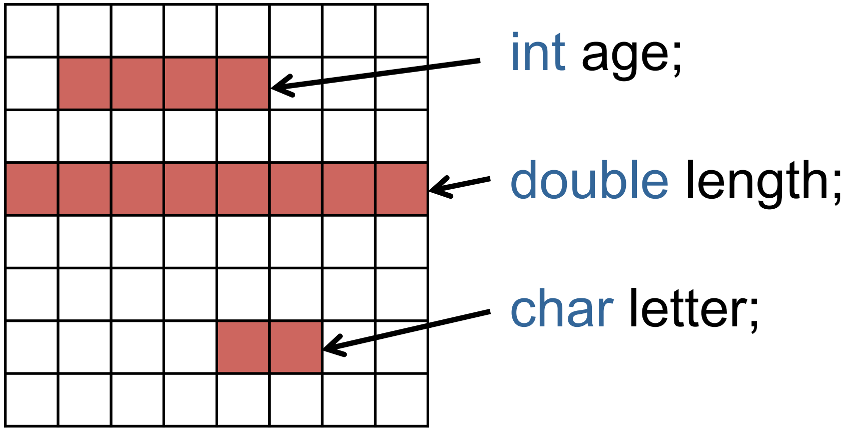
type *variable_name* = initial_value;

e.g.:

int i = 1;

boolean passedTest = true;

Polygon triangle = new Polygon();



main memory

Using Variables

- Specify the type only once at declaration
- In assignment operation, the right side is evaluated first. The value is then stored into the left side
- E.g.: Swap values of two integer variables

```
int a = 10;   int b = 5;
```

Method 1:

```
int c = a;
a = b;
b = c;
```

Method 2:

```
a = a + b;
b = a – b;
a = a – b;
```

Defined constants

- public static final *Type Variable* = *Constant*;
- Named in ALL_CAPS
- public class DefinedConstant

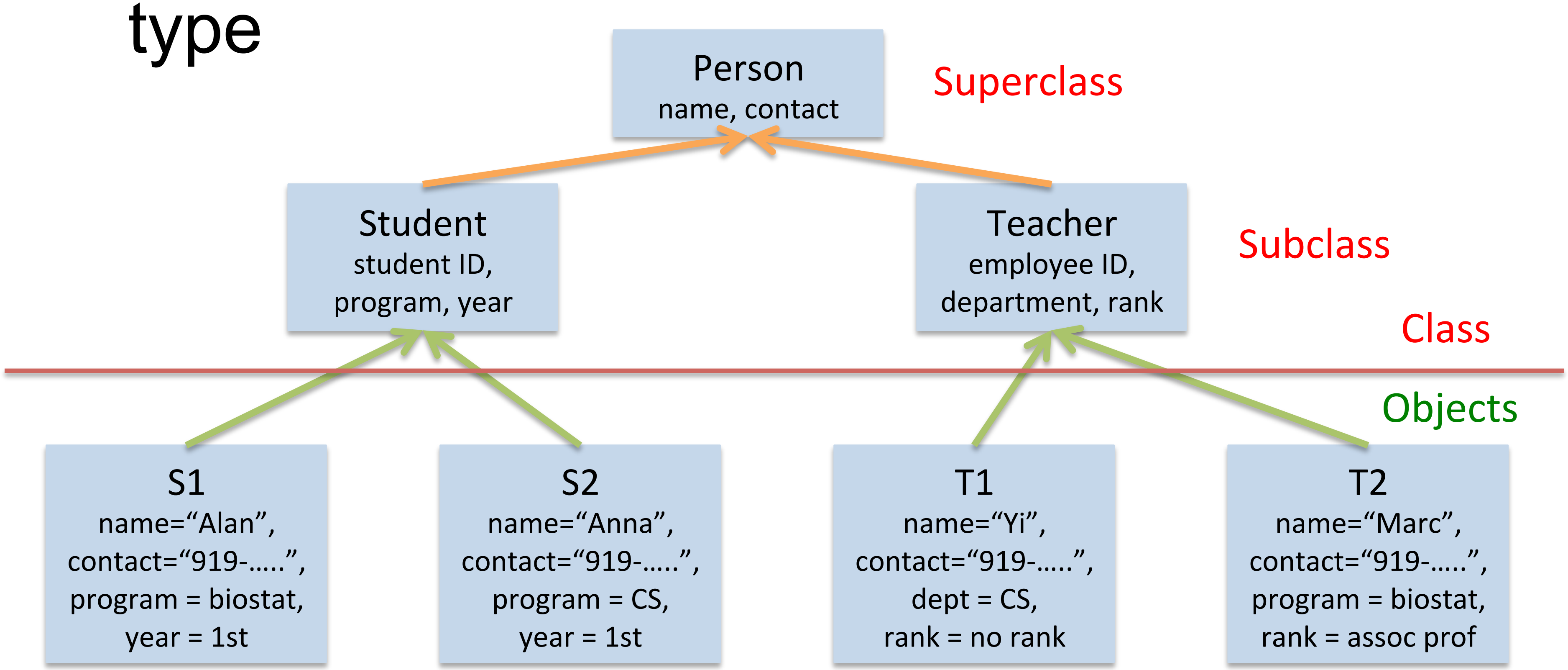
```
{
    public static final double PI = 3.14159;
    public static void main(String[] args){
    … }
}
```

Compatibility and Type Casting

- Variable of “Bigger” type can hold values of “Smaller” type
  - int a = 5;
  - double b = a;
- One can cast one type into another type (at the risk of losing information)
  - double a = 10.0 / 3.0; // a = 3.333333333…
  - int b = (int)a; // b = 3

Object Oriented Programming (OOP)

- Object: Attributes + Methods
- Class: the blueprint of objects of the same type



Java is an OOP language

- Encapsulation
  - “Information hiding”: putting things in a capsule
- Polymorphism
  - “Many forms”: the same instruction to mean the same thing in different contexts
- Inheritance
  - Organizing classes, so properties only have to be defined once

OOP in Practice

- Import class if necessary
  - E.g.: import java.util.*;
- Create object
  - Class_Type variable_name = new ClassType(…);
  - E.g.: Scanner keyboard = new Scanner(System.in);

    Polygon treeTop = new Polygon();
- Access object members (attribute or method)
  - int inputNumber = keyboard.nextInt();
  - treeTop.setColor(Color.green);

String

- A Class Type
- Objects of String class can be defined as:
  - String myString = “UNC is Great!”;
- Each String object consists of
  - A sequence of characters (char)

String	U	N	C		i	s		G	r	e	a	t	!
Indices:	0	1	2	3	4	5	6	7	8	9	10	11	12

  - A set of methods that can process the sequence of characters

String

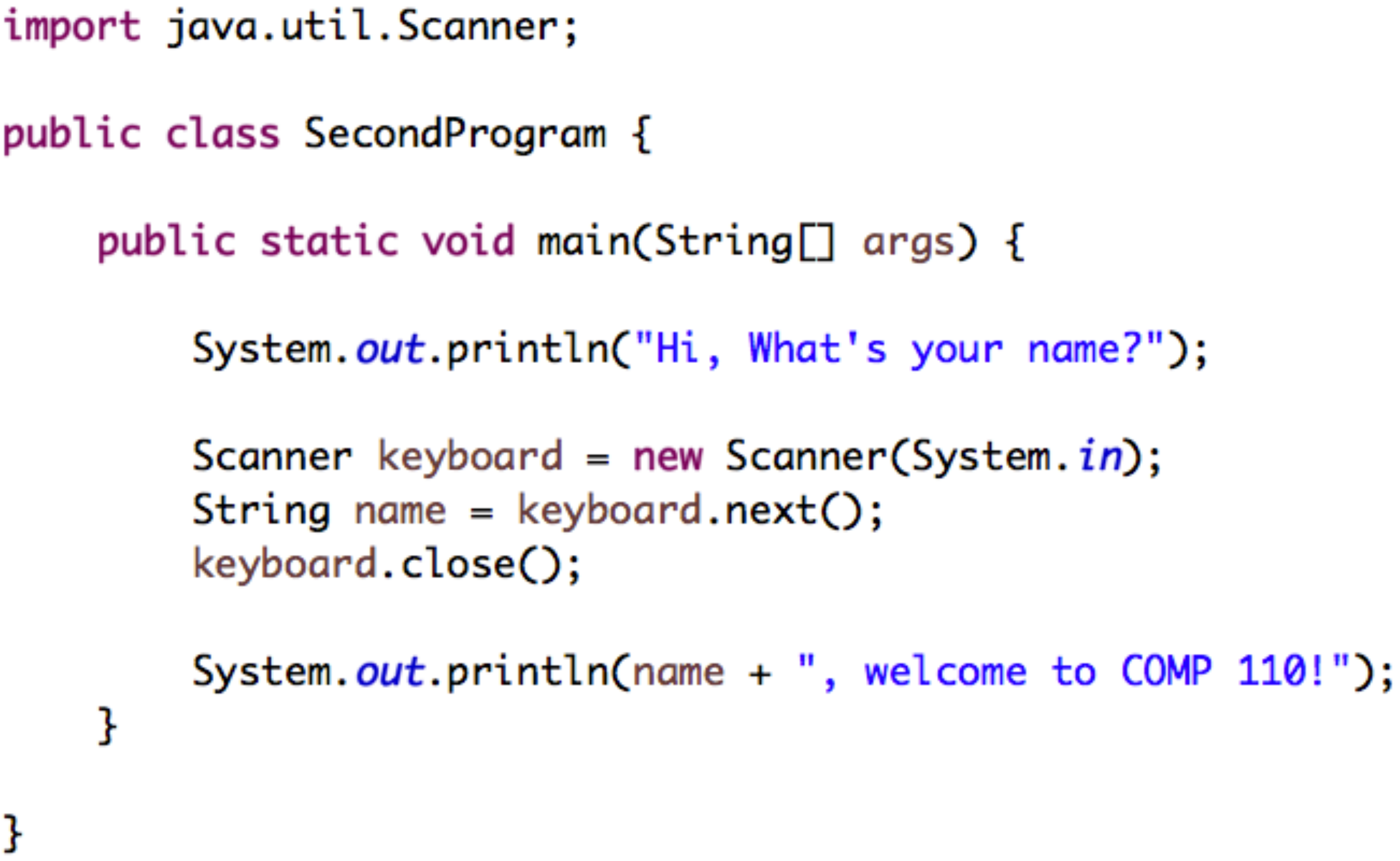
- Concatenation by "+"
  - "My name is " + "Yi" → "My name is Yi"
- Mixed operations
  - "The sum is " + 5 + 6 → "The sum is 56"
  - "The sum is " + (5+6) → "The sum is 11"
- More methods, see Java API for reference
  - length(), subString(), charAt(), toLowerCase() …

Console I/O

- Two built-in Java objects
  - System.in
  - System.out
- Console input: use Scanner class
  - Scanner keyboard = new Scanner(System.in);
  - int inputNumber = keyboard.nextInt();
  - String name = keyboard.next();
- Console output
  - System.out.print(…);
  - System.out.println(…);

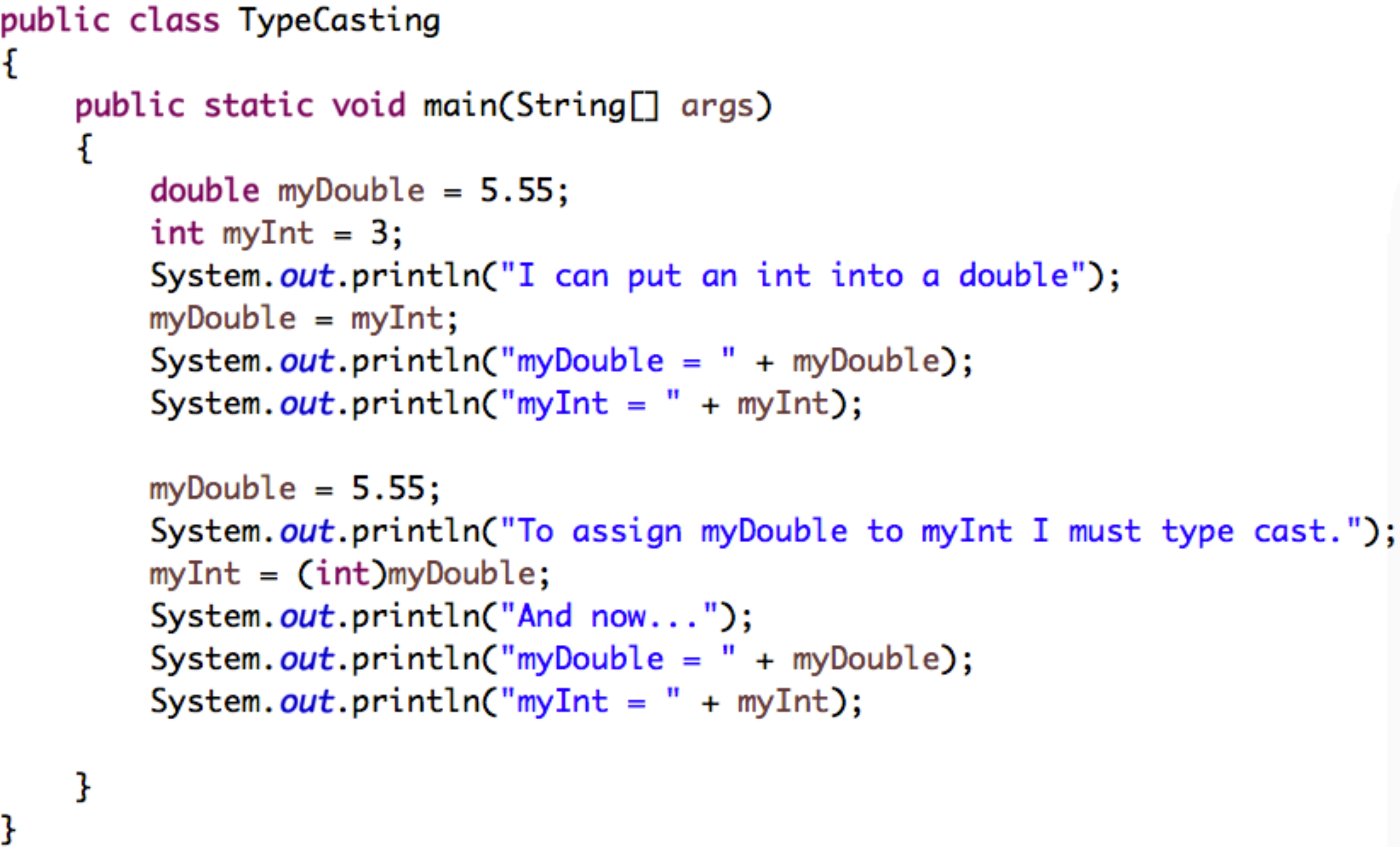
- Review Programs in Previous Lectures & Labs

SecondProgram.java in Lecture 2



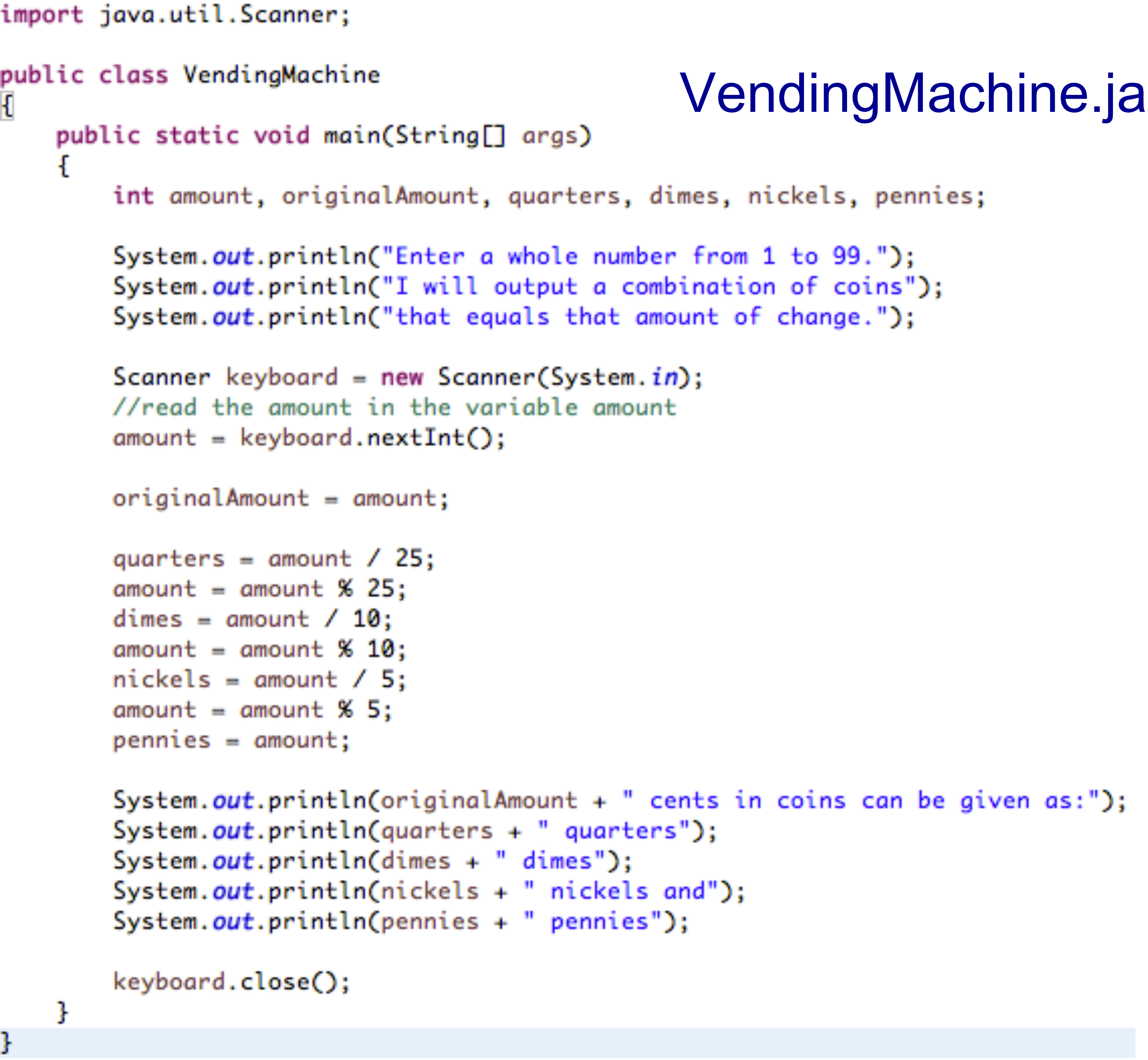
```
import java.util.Scanner;

public class SecondProgram {

    public static void main(String[] args) {

        System.out.println("Hi, What's your name?");

        Scanner keyboard = new Scanner(System.in);
        String name = keyboard.next();
        keyboard.close();

        System.out.println(name + ", welcome to COMP 110!");
    }

}
```

TypeCasting.java in Lecture 4



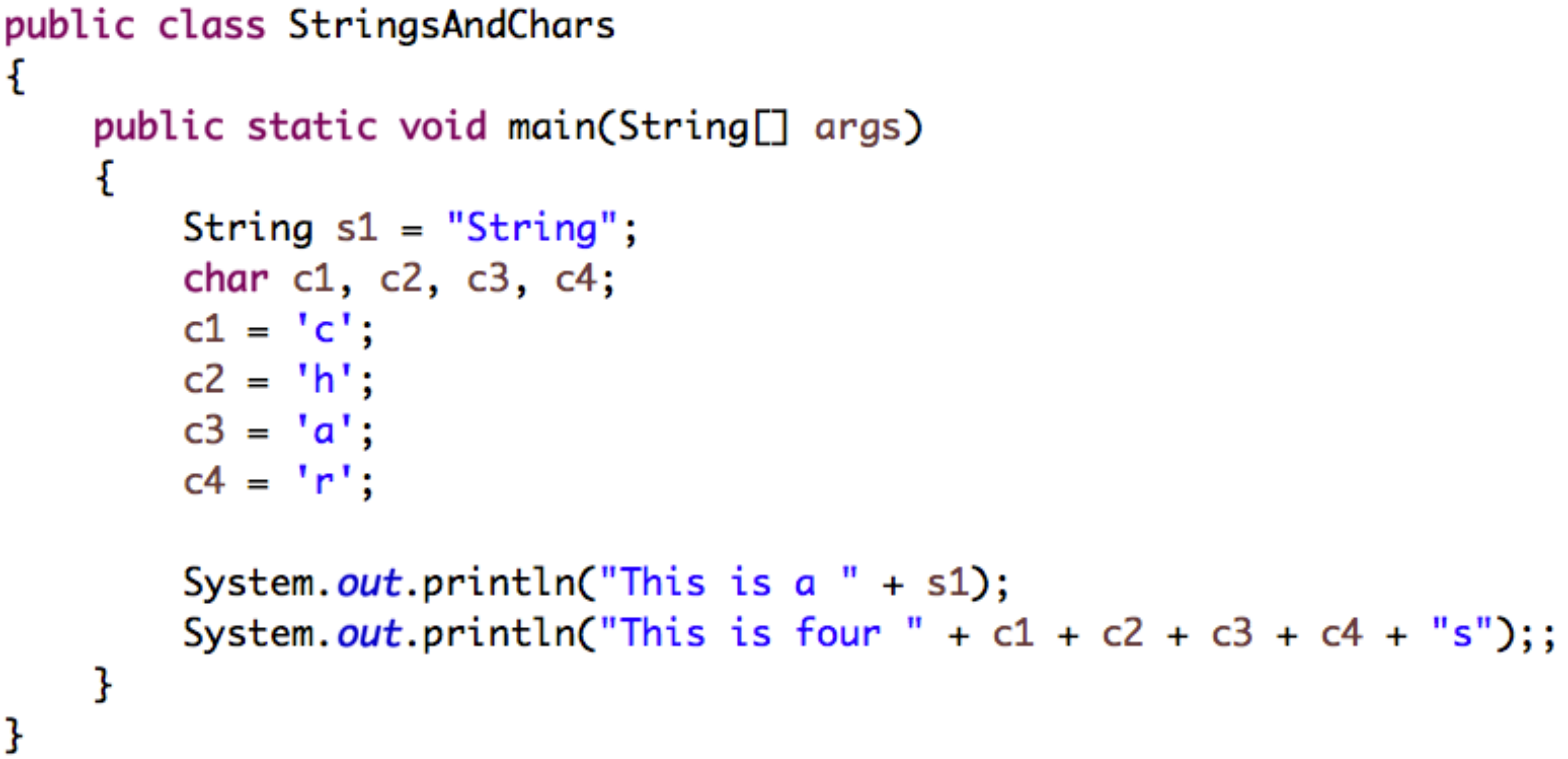
```
public class TypeCasting
{
    public static void main(String[] args)
    {
        double myDouble = 5.55;
        int myInt = 3;
        System.out.println("I can put an int into a double");
        myDouble = myInt;
        System.out.println("myDouble = " + myDouble);
        System.out.println("myInt = " + myInt);

        myDouble = 5.55;
        System.out.println("To assign myDouble to myInt I must type cast.");
        myInt = (int)myDouble;
        System.out.println("And now...");
        System.out.println("myDouble = " + myDouble);
        System.out.println("myInt = " + myInt);

    }
}
```

VendingMachine.java in Lab1



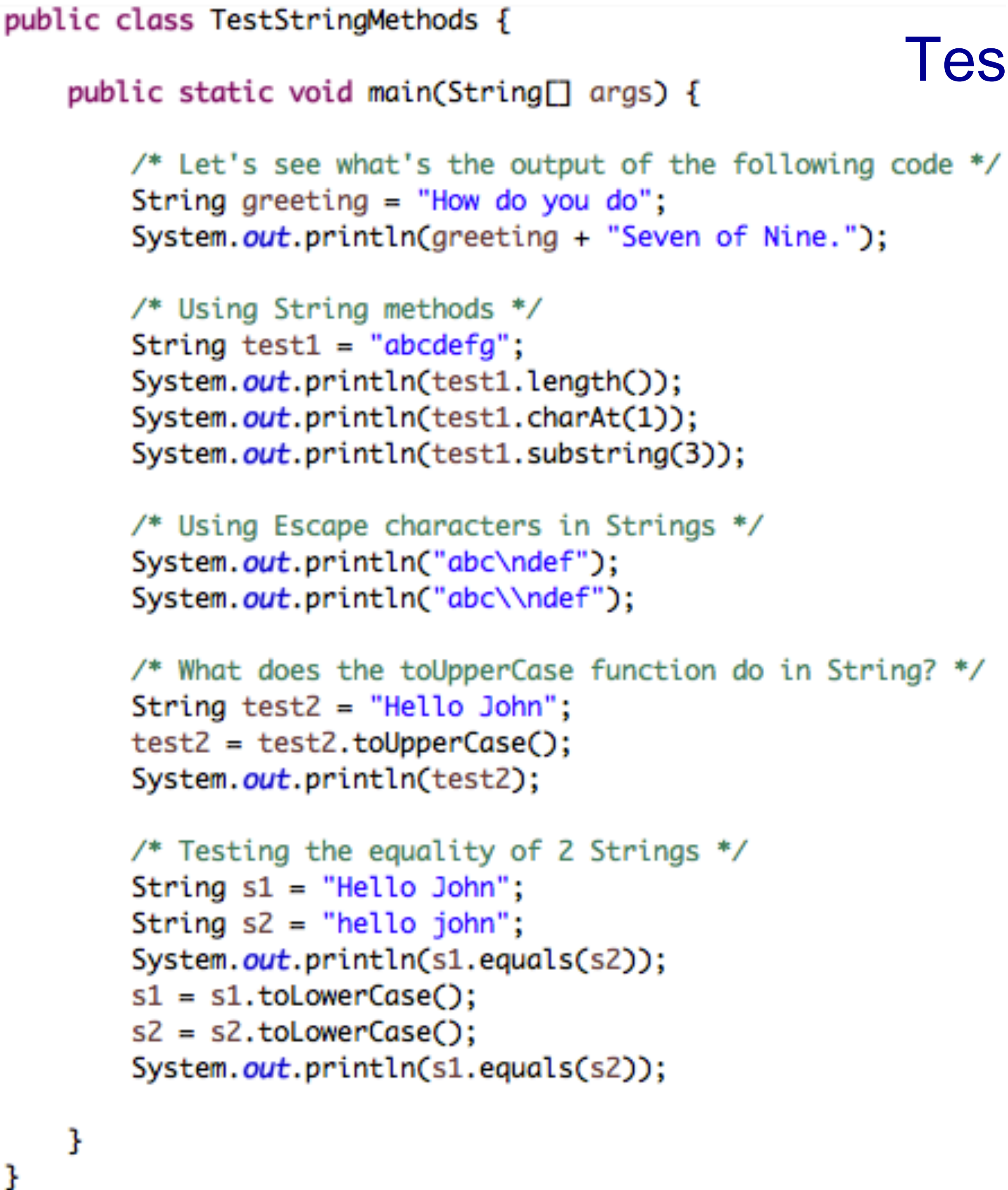
```java
import java.util.Scanner;

public class VendingMachine
{
    public static void main(String[] args)
    {
        int amount, originalAmount, quarters, dimes, nickels, pennies;

        System.out.println("Enter a whole number from 1 to 99.");
        System.out.println("I will output a combination of coins");
        System.out.println("that equals that amount of change.");

        Scanner keyboard = new Scanner(System.in);
        //read the amount in the variable amount
        amount = keyboard.nextInt();

        originalAmount = amount;

        quarters = amount / 25;
        amount = amount % 25;
        dimes = amount / 10;
        amount = amount % 10;
        nickels = amount / 5;
        amount = amount % 5;
        pennies = amount;

        System.out.println(originalAmount + " cents in coins can be given as:");
        System.out.println(quarters + " quarters");
        System.out.println(dimes + " dimes");
        System.out.println(nickels + " nickels and");
        System.out.println(pennies + " pennies");

        keyboard.close();
    }
}
```

StringsAndChars.java in Lecture 5



```java
public class StringsAndChars
{
    public static void main(String[] args)
    {
        String s1 = "String";
        char c1, c2, c3, c4;
        c1 = 'c';
        c2 = 'h';
        c3 = 'a';
        c4 = 'r';

        System.out.println("This is a " + s1);
        System.out.println("This is four " + c1 + c2 + c3 + c4 + "s");;
    }
}
```

TestStringMethods.java in Lecture 5



```java
public class TestStringMethods {

    public static void main(String[] args) {

        /* Let's see what's the output of the following code */
        String greeting = "How do you do";
        System.out.println(greeting + "Seven of Nine.");

        /* Using String methods */
        String test1 = "abcdefg";
        System.out.println(test1.length());
        System.out.println(test1.charAt(1));
        System.out.println(test1.substring(3));

        /* Using Escape characters in Strings */
        System.out.println("abc\ndef");
        System.out.println("abc\\ndef");

        /* What does the toUpperCase function do in String? */
        String test2 = "Hello John";
        test2 = test2.toUpperCase();
        System.out.println(test2);

        /* Testing the equality of 2 Strings */
        String s1 = "Hello John";
        String s2 = "hello john";
        System.out.println(s1.equals(s2));
        s1 = s1.toLowerCase();
        s2 = s2.toLowerCase();
        System.out.println(s1.equals(s2));

    }
}
```

Next Class

- Flow of control: Branching
- Reading assignments: Chapter 3.1-3.3